Fractal Decoded Image Quality Prediction

Wang Qiang[a*] and Du Jingjing[b]
[a]Dalian Maritime University, College of Information Science and Technology, Linghai Road, No.1, Dalian, China, 116026
[b]Hengshui University, College of Electronics and Information Engineering, Heping Road, No.1088, Hengshui, China, 053010
*Corresponding Author, E-mail: wangqiang2011@dlmu.edu.cn

Abstract: To predict fractal decoded image quality more efficiently, an effective decoded image quality prediction method was proposed in this study. In fractal encoding process, the dynamic range of the linear correlation coefficients (LCCs) between range blocks and their best-matched domain blocks was greatly extended by several outliers which increased uncertainty and resulted in reduced prediction accuracy. To remove the interference of outliers, we introduced the effective minimum and maximum of LCCs, which provided the effective bottom and top limits of the actual percentage of accumulated collage error (EBL-APACE and ETL-APACE), respectively. Further, when EBL-APACE reached a large percentage, the average collage error (ACER) can be estimated, and the decoded image quality can be predicted directly. Experimental results show that compared with the previous method, the proposed method can provide higher prediction accuracy with fewer computations.

Keywords: Fractal image coding; Decoded image quality; Accumulated collage error; Average collage error

1. Introduction

With the rapid development of modern society, people are gradually exposed to more and more images in their daily life, which brings great pressure to the transmission and storage of images. Efficient image compression technology is one of the important ways to solve the above problems. Different from conventional image compression techniques, fractal image coding has the characteristics of novel idea, potential high compression ratio, fast decoding, and resolution independence, which have attracted the attention of researchers worldwide[1-4]. However, fractal image coding has the drawback of high computational complexity in the encoding process, which seriously slows down its practical application. Thus, fast fractal image encoding becomes one of the key research directions. In recent years, many fast fractal encoding methods were proposed which can be mainly divided into two categories: 1) Local block-matching based fast fractal image encoding[5-7]. This type of methods accelerated the encoding process by excluding a number of domain blocks and converting global matching into local matching. 2) No-search fractal image coding[8-11]. This type of methods assigned the best-matched domain blocks directly without searching operations, and real-time encoding and higher compression ratio can be realized at the expense of decoded image quality. In addition to image compression, fractal image coding has been gradually applied to other image processing applications, such as image denoising[12-18] , image hashing[19,20], image retrieval[21-23], watermarking[24-26], image super resolution[27-31], head pose estimation[32, 33], and MRI image processing[34].

Predictable quality of decoded images is a novel property of fractal image coding. In previous work, the quality of decoded images was predicted only with partial encoding process and without decoding process[35]. In this study, an effective method was proposed to predict the quality of decoded images more accurately and quickly. First, we found that the dynamic range of the linear correlation coefficients between range blocks and their best-matched domain blocks was extended greatly by several outliers. Because the number of outliers was very small, they contributed little to the accumulated collage error (ACE) of all range blocks. Thus, we proposed a novel method to calculate the effective minimum and maximum of linear correlation coefficients to remove the interference of outliers, which provided the effective bottom and top limits of the actual percentage of accumulated collage error (EBL-APACE and ETL-APACE), respectively. Moreover, when EBL-APACE reached a large percentage (Such as 90%), EBL-APACE and ETL-APACE determined a small dynamic range for the actual percentage of accumulated collage error (APACE), and the average collage error (ACER) can be estimated approximately. Further, based on the logarithmic relationship between ACER and the fractal decoded image quality, the quality of decoded images can be predicted directly. Finally, we adopted four fractal encoding methods to assess the performance of the proposed method. Experimental results show that compared with the previous method, the proposed method can provide higher prediction accuracy with fewer computations.

This paper is organized as follows: Conventional fractal image coding is reviewed in Section 2. The principle of the proposed method is described in Section 3. In Section 4, the experimental results are presented and analyzed in detail. The final conclusion is given in Section 5.

2. Conventional fractal image coding

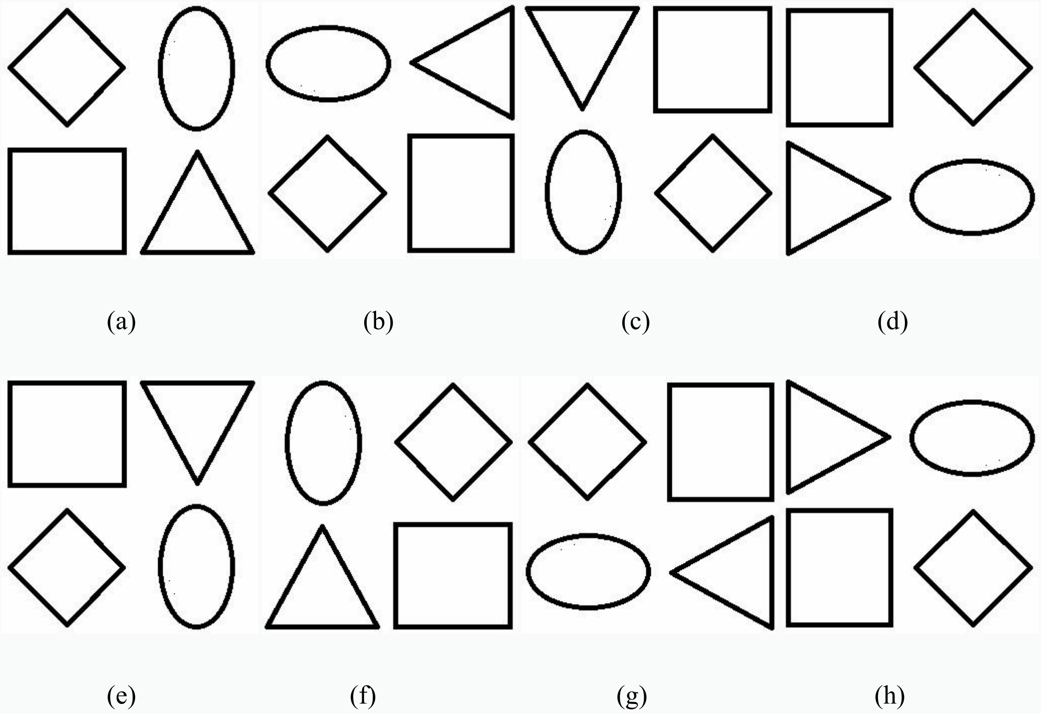
Fractal image coding aims to establish an iterated function system whose fixed point can approach the original image approximately. The first block-based fractal encoding method was proposed by Jacquin[1]. First, the original $M \times N$ image $\boldsymbol{f}$ was uniformly divided into nonoverlapping $B \times B$ range blocks $\boldsymbol{R}_i$, i=1,2,3,...,NumR, where NumR denotes the total number of range blocks. Then, domain blocks $\boldsymbol{D}_j$, j=1,2,3,...,NumD, can be obtained by sliding a $2B \times 2B$ window over the original image from left to right and from top to bottom, where NumD denotes the total number of domain blocks. Further, domain blocks were contracted to the same size of range blocks, and an extended domain block pool (EDBP) was obtained by performing eight isometric transformations on domain blocks. For a given image, Figure 1 illustrates the eight isometric transformations which contain rotation through 0 degree, 90 degree, 180 degree, 270 degree, reflection about the middle horizontal axis, reflection about the middle vertical axis, reflection about the main diagonal and reflection about the secondary diagonal in order. Finally, the best-matched domain block can be obtained by minimizing the following function

$$\begin{gathered}\mathrm{CE}(\boldsymbol{R}_i) = \min_{\boldsymbol{D}_j}\left(\min_{s_i, o_i} \frac{1}{B^2}\left\|\boldsymbol{R}_i - s_i\boldsymbol{D}_j - o_i\boldsymbol{I}\right\|^2\right) \\ i = 1,2,3,\ldots,\mathrm{NumR},\ j = 1,2,3,\ldots,\mathrm{NumD}\times 8\end{gathered} \tag{1}$$

where CE($\boldsymbol{R}_i$) denotes the collage error of $\boldsymbol{R}_i$. $\boldsymbol{I}$ denotes a $B \times B$ matrix whose elements are all ones. s_i and o_i denote scaling and offset coefficients, respectively. $\boldsymbol{D}_j$ is the best-matched domain block of $\boldsymbol{R}_i$. With the least square method, s_i and o_i can be calculated as follows

$$s_i = \left\langle \boldsymbol{R}_i - \overline{r}_i\boldsymbol{I},\quad \boldsymbol{D}_j - \overline{d}_j\boldsymbol{I}\right\rangle \Big/ \left\|\boldsymbol{D}_j - \overline{d}_j\boldsymbol{I}\right\|^2,\quad o_i = \overline{r}_i - s_i\overline{d}_j \tag{2}$$

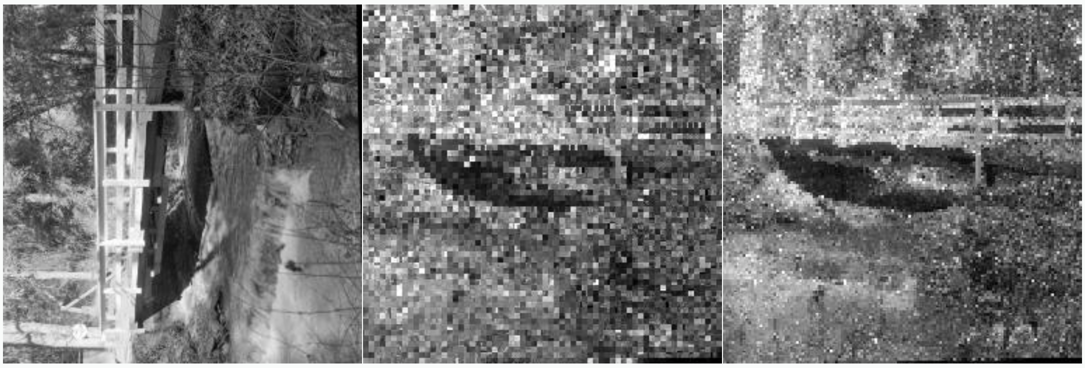
where $\langle \bullet,\bullet \rangle$ denotes the inner product. $\overline{r}_i$ and $\overline{d}_j$ are the averages of $\boldsymbol{R}_i$ and $\boldsymbol{D}_j$, respectively.

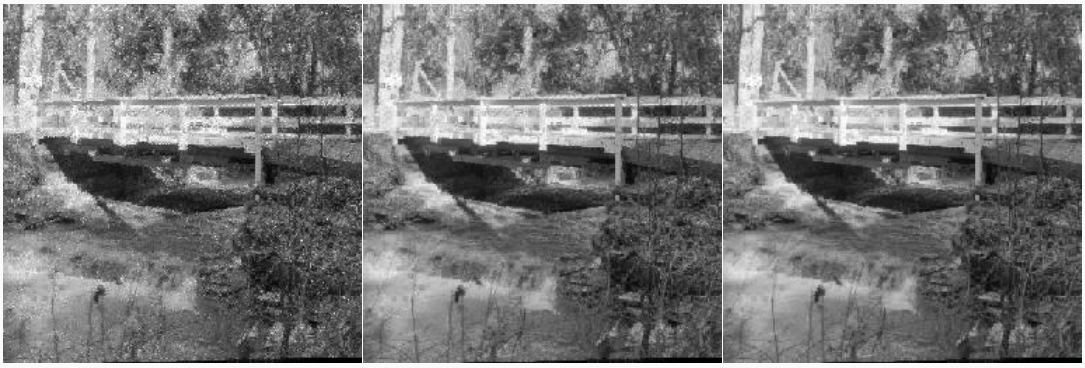

(a) (b) (c) (d)

(e) (f) (g) (h)

Fig.1 Eight isometric transformations.

In decoding process, arbitrary $M \times N$ image can be selected as the initial image. First, the same transformations as those in encoding process were conducted from the best-matched domain blocks to corresponding range blocks. One iteration can be completed after all range blocks were reconstructed. After about 10 iterations, the decoded image can be obtained. Figure 2 (a) selects a rotated 256 × 256 Bridge as the initial image. When we encoded Bridge in the encoding process, Figure 3 (b) to (f) illustrate the first five iteration images in decoding process. We see that the iteration images converge to Bridge gradually.


(a) Initial Image (b) First iteration (c) Second iteration



(d) Third iteration (e) Forth iteration (f) Fifth iteration

Fig.2 Fractal decoding process.

3. Proposed method

3.1 Effective minimum and maximum of the linear correlation coefficients between range blocks and their best-matched domain blocks

For $\boldsymbol{R}_i$, substituting (2) back to (1) yields

$$\begin{aligned}\mathrm{CE}(\boldsymbol{R}_i) &= \frac{1}{B^2}\left\|\boldsymbol{R}_i - s_i\boldsymbol{D}_j - o_i\boldsymbol{I}\right\|^2 \\ &= \frac{1}{B^2}\left(\left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2 - s_i^2\left\|\boldsymbol{D}_j - \bar{d}_j\boldsymbol{I}\right\|^2\right) \\ &= \frac{1}{B^2}\left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left(1 - \frac{\left|\left\langle \boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}, \boldsymbol{D}_j - \bar{d}_j\boldsymbol{I}\right\rangle\right|^2}{\left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left\|\boldsymbol{D}_j - \bar{d}_j\boldsymbol{I}\right\|^2}\right) \\ &= \frac{1}{B^2}\left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left(1 - \mathrm{LCC}_i^2\right)\end{aligned} \tag{3}$$

where LCC_i denotes the linear correlation coefficient between $\boldsymbol{R}_i$ and $\boldsymbol{D}_j$. Because LCC_i^2 lies between 0 and 1, Eq. (3) satisfies

$$\mathrm{CE}(\boldsymbol{R}_i) \le \frac{1}{B^2}\left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2 \tag{4}$$

In Eq. (4), the variance of $\boldsymbol{R}_i$ provides the top limit of CE($\boldsymbol{R}_i$), and thus only the range blocks with large variances may result in large collage errors. By Eq. (3), ACE of all range blocks can be calculated as

$$\mathrm{ACE}^{\mathrm{All}} = \sum_{i=1}^{\mathrm{NumR}} \mathrm{CE}(\boldsymbol{R}_i) = \frac{1}{B^2}\sum_{i=1}^{\mathrm{NumR}} \left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left[1 - \left(\mathrm{LCC}_i\right)^2\right] \tag{5}$$

If we use $\mathrm{LCC}_{\mathrm{Max}}$ and $\mathrm{LCC}_{\mathrm{Min}}$ to denote the maximum and minimum of LCC_i, i=1,2,3,...,NumR, respectively, and by Eq. (5), we have

$$\mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min}} < \mathrm{ACE}^{\mathrm{All}} < \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Max}} \tag{6}$$

where

$$\begin{cases} \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min}} = \sum_{i=1}^{\mathrm{NumR}} \mathrm{CE}(\boldsymbol{R}_i)_{\mathrm{Min}} = \dfrac{1}{B^2} \sum_{i=1}^{\mathrm{NumR}} \left\| \boldsymbol{R}_i - \bar{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\mathrm{LCC}_{\mathrm{Max}} \right)^2 \right] \\ \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Max}} = \sum_{i=1}^{\mathrm{NumR}} \mathrm{CE}(\boldsymbol{R}_i)_{\mathrm{Max}} = \dfrac{1}{B^2} \sum_{i=1}^{\mathrm{NumR}} \left\| \boldsymbol{R}_i - \bar{r}_i \boldsymbol{I} \right\|^2 \left[1 - \left(\mathrm{LCC}_{\mathrm{Min}} \right)^2 \right] \end{cases} \tag{7}$$

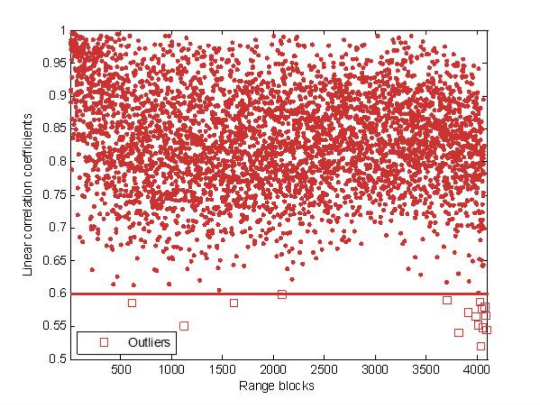


Fig.3 Distribution of linear correlation coefficients.

In encoding process, we sorted the range blocks by their variances from largest to smallest and encoded them in order. For Bridge, Figure 3 shows the distribution of LCCs. In Eq. (7), $\mathrm{LCC}_{\mathrm{Max}}$ and $\mathrm{LCC}_{\mathrm{Min}}$ provide the bottom and top limits of $\mathrm{ACE}^{\mathrm{All}}$, $\mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min}}$ and $\mathrm{ACE}^{\mathrm{All}}_{\mathrm{Max}}$, respectively. However, we observe that the dynamic range of LCCs is mainly determined by several outliers which greatly extend the dynamic range of LCCs. In Fig.3, all the LCCs below the red line, i.e., smaller than 0.6, are labeled by “□”. The outliers make $\mathrm{LCC}_{\mathrm{Min}}$ extend to below 0.6. Because the outliers’ ACE only comprise 1.47% of total collage errors, their effect can be ignored. To remove the interference of outliers and effectively estimate the dynamic range of $\mathrm{ACE}^{\mathrm{All}}$, we adopt the following method to compute the effective minimum and maximum of LCCs:

For all range blocks, we established a histogram of collage errors with respect to LCCs. If we had NumC range blocks $\boldsymbol{R}_k$, k=1,2,3,...,NumC, whose LCCs were all lccs. We used p(lcc) to represent the normalized version of the sum of the collage errors with the same LCC as

$$p(\mathrm{lcc})=\sum_{k=1}^{\mathrm{NumC}}\mathrm{CE}_k(\mathrm{lcc})\Big/\mathrm{ACE}^{\mathrm{All}} \qquad \mathrm{lcc}\in[0,1] \tag{8}$$

where $\mathrm{CE}_k(\mathrm{lcc})$ denotes the collage error of $\boldsymbol{R}_k$ with LCC=lcc. From Eq. (8), we know that $p(\mathrm{lcc})$ actually represents the probability density function of collage errors (PDF-CE) with respect to LCCs.

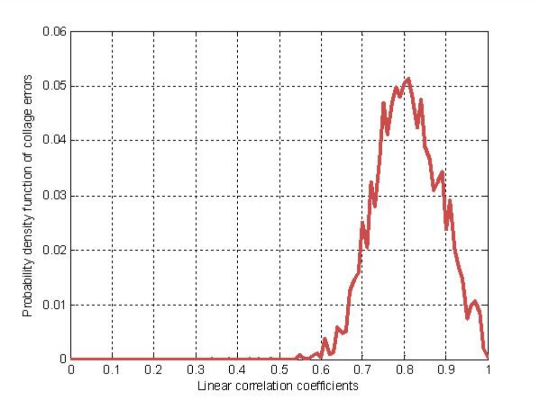
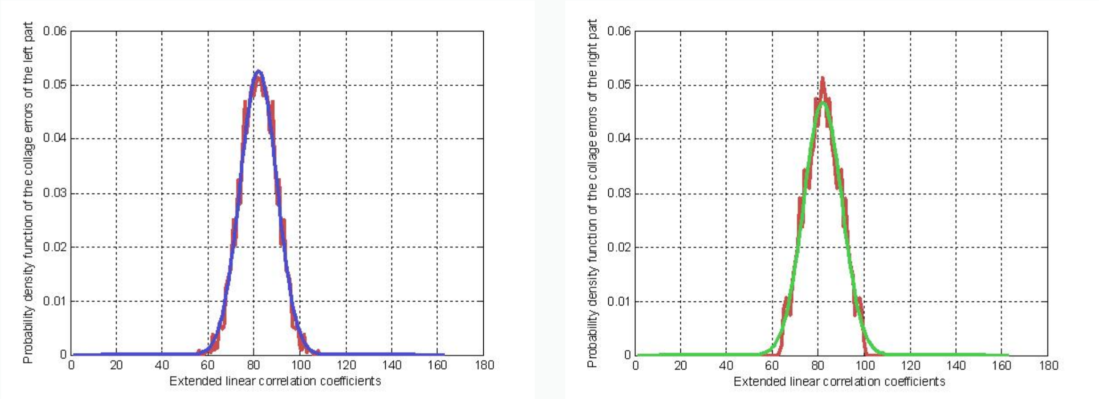


Fig.4 PDF-CE versus LCCs.



(a) Fitting result of the left part and its reflection.(b) Fitting result of the left part and its reflection.

Fig.5 Fitting results for the left and right parts of PDF-CE in Fig.5, respectively.

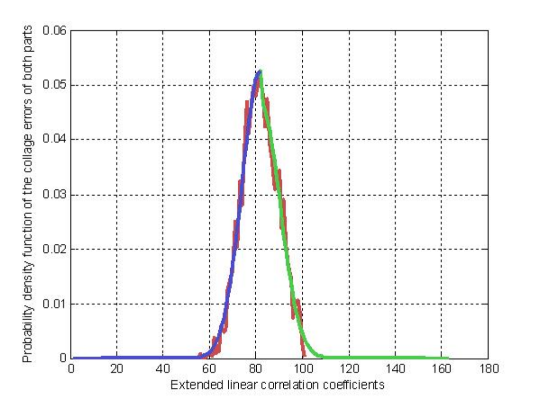


Fig.6 Fitting results of PDF-CE.

Figure 4 shows PDF-CE for Bridge which is asymmetric. We plotted a vertical line (VL) at lcc=T where PDE-CE reached its peak value. VL divides PDF-CE into the left and right parts which comprise T_{Left}% and T_{Right}% of ACE$^{\text{All}}$, respectively, and we have

$$T_{\text{Left}}\%+T_{\text{Right}}\%=100\% \tag{9}$$

We took out the left part and reflect it about VL, a symmetric curve in Fig.5 (a) was obtained. Further, by reflecting the right part about VL, we also obtained a symmetric curve in Fig.5 (b). Then, we adopted the following two Gaussian functions, GF$_{\text{Left}}$(lcc) and GF$_{\text{Right}}$(lcc), to fit the two curves in Fig.5 (a) and (b), respectively.

$$\begin{cases} \text{GF}_{\text{Left}}\left(\text{lcc}\right)=\alpha_1 e^{-\frac{(\text{lcc}-\alpha_2)^2}{2\alpha_3^2}} \\ \text{GF}_{\text{Right}}\left(\text{lcc}\right)=\beta_1 e^{-\frac{(\text{lcc}-\beta_2)^2}{2\beta_3^2}} \end{cases} \tag{10}$$

where a_3 and β_3 represent the standard deviations of the curves in Fig.5 (a) and (b), respectively. a_2 and β_2 are both equal to T. We know that Gaussian distributions concentrate 95.5% × 2T_{Left}% and 95.5% × 2T_{Right}% of ACE$^{\text{All}}$ in the intervals of [a_2-2a_3, a_2+2a_3] and [β_2-2β_3, β_2+2β_3] in Fig.5 (a) and (b), respectively. Thus, [a_2-2a_3, a_2] and [β_2, β_2+2β_3] concentrate 95.5% × T_{Left}% and 95.5% × T_{Right}% of ACE$^{\text{All}}$, respectively. Further, we took out the left and right parts of the fitting curve in Fig.5 (a) and (b), respectively, and combined them together in Fig.6. By Eq. (9), we know that the interval [a_2-2a_3, β_2+2β_3] concentrate 95.5% of ACE$^{\text{All}}$. Finally, we define a_2-2a_3 and β_2+2β_3 as the effective minimum and maximum of LCCs, LCC$_{\text{Min\_Effective}}$ and LCC$_{\text{Max\_Effective}}$, respectively, as

$$\begin{cases} \text{LCC}_{\text{Min\_Effective}}=\alpha_2-2\alpha_3 \\ \text{LCC}_{\text{Max\_Effective}}=\beta_2+2\beta_3 \end{cases} \tag{11}$$

LCC$_{\text{Min\_Effective}}$ and LCC$_{\text{Max\_Effective}}$ can effectively remove the effect of outliers. Because the the dynamic range of LCCs is narrowed, we also have the inequalities as

$$\begin{cases} \text{LCC}_{\text{Min\_Effective}}>\text{LCC}_{\text{Min}} \\ \text{LCC}_{\text{Max\_Effective}}<\text{LCC}_{\text{Max}} \end{cases} \tag{12}$$

3.2 Effective bottom and top limits of accumulated collage error

With LCC$_{\text{Min\_Effective}}$ and LCC$_{\text{Max\_Effective}}$, Eq. (7) can be modified as

$$\begin{cases} \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min\_Effective}} = \sum_{i=1}^{\mathrm{NumR}} \mathrm{CE}(\boldsymbol{R}_i)_{\mathrm{Min\_Effective}} = \frac{1}{B^2}\sum_{i=1}^{\mathrm{NumR}} \left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}_{\mathrm{Max\_Effective}}\right)^2\right] \\ \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Max\_Effective}} = \sum_{i=1}^{\mathrm{NumR}} \mathrm{CE}(\boldsymbol{R}_i)_{\mathrm{Max\_Effective}} = \frac{1}{B^2}\sum_{i=1}^{\mathrm{NumR}} \left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}_{\mathrm{Min\_Effective}}\right)^2\right] \end{cases} \tag{13}$$

We compared $\mathrm{ACE}^{\mathrm{All}}$ with $\mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min\_Effective}}$ as

$$\begin{aligned} \mathrm{ACE}^{\mathrm{All}} - \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min\_Effective}} &= \frac{1}{B^2}\sum_{i=1}^{\mathrm{NumR}}\left[\mathrm{CE}(\boldsymbol{R}_i) - \mathrm{CE}(\boldsymbol{R}_i)_{\mathrm{Min\_Effective}}\right] \\ &= \frac{1}{B^2}\sum_{i=1}^{\mathrm{NumR}} \left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left[\left(\mathrm{LCC}_{\mathrm{Max\_Effective}}\right)^2 - \left(\mathrm{LCC}_i\right)^2\right] \end{aligned} \tag{14}$$

The range blocks can be divided into two categories: Range blocks whose LCCs are above and below $\mathrm{LCC}_{\mathrm{Max\_Effective}}$, respectively. We use $\mathrm{Num}^{\mathrm{Above}}$ and $\mathrm{Num}^{\mathrm{Below}}$ to denote the numbers of range blocks whose LCCs are above and below $\mathrm{LCC}_{\mathrm{Max\_Effective}}$, respectively. Then, Eq. (14) can be rewritten as

$$\begin{aligned} \mathrm{ACE}^{\mathrm{All}} - \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min\_Effective}} = &\underbrace{\frac{1}{B^2}\sum_{i=1}^{\mathrm{Num}^{\mathrm{Above}}} \left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left[\left(\mathrm{LCC}_{\mathrm{Max\_Effective}}\right)^2 - \left(\mathrm{LCC}_i\right)^2\right]}_{<0} + \\ &\underbrace{\frac{1}{B^2}\sum_{i=1}^{\mathrm{Num}^{\mathrm{Below}}} \left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left[\left(\mathrm{LCC}_{\mathrm{Max\_Effective}}\right)^2 - \left(\mathrm{LCC}_i\right)^2\right]}_{>0} \end{aligned} \tag{15}$$

Only the LCCs of several outliers satisfy $\mathrm{LCC}_i > \mathrm{LCC}_{\mathrm{Max\_Effective}}$, and the others satisfy $\mathrm{LCC}_i < \mathrm{LCC}_{\mathrm{Max\_Effective}}$. For Bridge, $\mathrm{Num}^{\mathrm{Above}}=39$, $\mathrm{Num}^{\mathrm{Below}}=4{,}057$, and $\mathrm{ACE}^{\mathrm{All}} - \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min\_Effective}} = 529{,}480 > 0$. Generally, the number of LCCs below $\mathrm{LCC}_{\mathrm{Max\_Effective}}$ is more than ten times larger than above $\mathrm{LCC}_{\mathrm{Max\_Effective}}$. Thus, in Eq. (15), in the case of $\mathrm{Num}^{\mathrm{Below}} >> \mathrm{Num}^{\mathrm{Above}}$, $\mathrm{ACE}^{\mathrm{All}} - \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min\_Effective}} > 0$, and we have

$$\mathrm{ACE}^{\mathrm{All}} > \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Min\_Effective}} \tag{16}$$

Then, we compared $\mathrm{ACE}^{\mathrm{All}}$ with $\mathrm{ACE}^{\mathrm{All}}_{\mathrm{Max\_Effective}}$ as

$$\begin{aligned} \mathrm{ACE}^{\mathrm{All}}_{\mathrm{Max\_Effective}} - \mathrm{ACE}^{\mathrm{All}} &= \frac{1}{B^2}\sum_{i=1}^{\mathrm{NumR}}\left[\mathrm{CE}(\boldsymbol{R}_i)_{\mathrm{Max\_Effective}} - \mathrm{CE}(\boldsymbol{R}_i)\right] \\ &= \frac{1}{B^2}\sum_{i=1}^{\mathrm{NumR}} \left\|\boldsymbol{R}_i - \bar{r}_i\boldsymbol{I}\right\|^2\left[\left(\mathrm{LCC}_i\right)^2 - \left(\mathrm{LCC}_{\mathrm{Min\_Effective}}\right)^2\right] \end{aligned} \tag{17}$$

Similar with Eq. (15), we divided range blocks into two categories: The range blocks whose

LCCs are above and below LCC$_{\text{Min\_Fitting}}$, respectively. We use Num$^{\text{Above}}$ and Num$^{\text{Below}}$ to denote the numbers of range blocks whose LCCs are above and below LCC$_{\text{Min\_Effective}}$, respectively. Then, Eq. (17) can be rewritten as

$$\begin{aligned}\text{ACE}^{\text{All}}_{\text{Max\_Effective}}-\text{ACE}^{\text{All}}=&\underbrace{\frac{1}{B^2}\sum_{i=1}^{\text{Num}^{\text{Above}}}\left\|\boldsymbol{R}_i-\bar{r}_i\boldsymbol{I}\right\|^2\left[\left(\text{LCC}_i\right)^2-\left(\text{LCC}_{\text{Min\_Effective}}\right)^2\right]}_{>0}+\\&\underbrace{\frac{1}{B^2}\sum_{i=1}^{\text{Num}^{\text{Bebow}}}\left\|\boldsymbol{R}_i-\bar{r}_i\boldsymbol{I}\right\|^2\left[\left(\text{LCC}_i\right)^2-\left(\text{LCC}_{\text{Min\_Effective}}\right)^2\right]}_{<0}\end{aligned}\tag{18}$$

Only the LCCs of several outliers satisfy LCC$_i$<LCC$_{\text{Min\_Effective}}$, and the others satisfy LCC$_i$>LCC$_{\text{Min\_Effective}}$, For Bridge, Num$^{\text{Above}}$=4,026, Num$^{\text{Below}}$=70, and $\text{ACE}^{\text{All}}_{\text{Max\_Effective}}-\text{ACE}^{\text{All}}=760,276>0$. Thus, in Eq. (18), in the case of Num$^{\text{Above}}$>>Num$^{\text{Below}}$, we have

$$\text{ACE}^{\text{All}}_{\text{Max\_Effective}}>\text{ACE}^{\text{All}}\tag{19}$$

By Eqs. (16) and (19), we have

$$\text{ACE}^{\text{All}}_{\text{Min\_Effective}}<\text{ACE}^{\text{All}}<\text{ACE}^{\text{All}}_{\text{Max\_Effective}}\tag{20}$$

Thus, the effective minimum and maximum of LCCs, LCC$_{\text{Min\_Effective}}$ and LCC$_{\text{Max\_Effective}}$, can provide the effective top and bottom limits of ACE$^{\text{All}}$, $\text{ACE}^{\text{All}}_{\text{Max\_Effective}}$ and $\text{ACE}^{\text{All}}_{\text{Min\_Effective}}$, respectively.

3.3 Decoded image quality prediction

In encoding process, range blocks were divided into two categories: Coded range blocks and uncoded range blocks. For coded ones, if the number of them is large enough, we can use the effective minimum and maximum of the LCCs of coded range blocks, $\text{LCC}^{\text{Coded}}_{\text{Min\_Effective}}$ and $\text{LCC}^{\text{Coded}}_{\text{Max\_Effective}}$, to approximate those of all range blocks, LCC$_{\text{Min\_Effective}}$ and LCC$_{\text{Max\_Effective}}$, as

$$\text{LCC}^{\text{Coded}}_{\text{Min\_Effective}}\approx\text{LCC}_{\text{Min\_Effective}},\quad\text{LCC}^{\text{Coded}}_{\text{Max\_Effective}}\approx\text{LCC}_{\text{Max\_Effective}}\tag{21}$$

Moreover, the effective minimum and maximum of the LCCs of uncoded range blocks, $\text{LCC}^{\text{Uncoded}}_{\text{Min\_Effective}}$ and $\text{LCC}^{\text{Uncoded}}_{\text{Max\_Effective}}$, can be represented by those of all range blocks as

$$\mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Min\_Effective}}=\mathrm{LCC}_{\mathrm{Min\_Effective}},\quad \mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Max\_Effective}}=\mathrm{LCC}_{\mathrm{Max\_Effective}} \tag{22}$$

By Eqs. (21) and (22), we can use $\mathrm{LCC}^{\mathrm{Coded}}_{\mathrm{Min\_Effective}}$ and $\mathrm{LCC}^{\mathrm{Coded}}_{\mathrm{Max\_Effective}}$ to approximate $\mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Min\_Effective}}$ and $\mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Max\_Effective}}$ as

$$\mathrm{LCC}^{\mathrm{Coded}}_{\mathrm{Min\_Effective}} \approx \mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Min\_Effective}},\quad \mathrm{LCC}^{\mathrm{Coded}}_{\mathrm{Max\_Effective}} \approx \mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Max\_Effective}} \tag{23}$$

Further, by Eq. (3), the collage error of arbitrary uncoded range block can be denoted as

$$\mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}}) = \frac{1}{B^2}\left\|\boldsymbol{R}_i^{\mathrm{Uncoded}} - \overline{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}_i^{\mathrm{Uncoded}}\right)^2\right], i=1,2,3,\ldots,\mathrm{Num}^{\mathrm{Uncoded}} \tag{24}$$

where $\mathrm{Num}^{\mathrm{Uncoded}}$ denotes the number of uncoded range blocks. $\mathrm{LCC}_i^{\mathrm{Uncoded}}$ denotes the LCC of $\boldsymbol{R}_i^{\mathrm{Uncoded}}$. Then, the ACE of uncoded range blocks can be described as

$$\mathrm{ACE}^{\mathrm{Uncoded}} = \sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}}) = \frac{1}{B^2}\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \left\|\boldsymbol{R}_i^{\mathrm{Uncoded}} - \overline{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}_i^{\mathrm{Uncoded}}\right)^2\right] \tag{25}$$

We assume that the result in Subsection 3.2 is also suitable for the uncoded range blocks, i.e., the effective minimum and maximum of the LCCs of uncoded range blocks, $\mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Min\_Effective}}$ and $\mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Max\_Effective}}$, can provide the effective top and bottom limits for $\mathrm{ACE}^{\mathrm{Uncoded}}$ as

$$\mathrm{ACE}^{\mathrm{Uncoded}}_{\mathrm{Min\_Effective}} < \mathrm{ACE}^{\mathrm{Uncoded}} < \mathrm{ACE}^{\mathrm{Uncoded}}_{\mathrm{Max\_Effective}} \tag{26}$$

where

$$\begin{cases} \mathrm{ACE}^{\mathrm{Uncoded}}_{\mathrm{Min\_Effective}} = \displaystyle\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Min\_Effective}} = \frac{1}{B^2}\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \left\|\boldsymbol{R}_i^{\mathrm{Uncoded}} - \overline{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Max\_Effective}}\right)^2\right] \\ \mathrm{ACE}^{\mathrm{Uncoded}}_{\mathrm{Max\_Effective}} = \displaystyle\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Max\_Effective}} = \frac{1}{B^2}\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \left\|\boldsymbol{R}_i^{\mathrm{Un\ coded}} - \overline{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}^{\mathrm{Uncoded}}_{\mathrm{Min\_Effective}}\right)^2\right] \end{cases} \tag{27}$$

By Eq. (23), Eq. (27) can be rewritten as

$$\begin{cases} \mathrm{ACE}^{\mathrm{Uncoded}}_{\mathrm{Min\_Effective}} = \displaystyle\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Min\_Effective}} = \frac{1}{B^2}\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \left\|\boldsymbol{R}_i^{\mathrm{Uncoded}} - \overline{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}^{\mathrm{Coded}}_{\mathrm{Max\_Effective}}\right)^2\right] \\ \mathrm{ACE}^{\mathrm{Uncoded}}_{\mathrm{Max\_Effective}} = \displaystyle\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Max\_Effective}} = \frac{1}{B^2}\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}} \left\|\boldsymbol{R}_i^{\mathrm{Uncoded}} - \overline{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}^{\mathrm{Coded}}_{\mathrm{Min\_Effective}}\right)^2\right] \end{cases} \tag{28}$$

Moreover, if the actual percentage of accumulated collage error (APACE) is defined as the ratio of $\mathrm{ACE}^{\mathrm{Coded}}$ to $\mathrm{ACE}^{\mathrm{All}}$ as

$$\mathrm{APACE}=\frac{\mathrm{ACE}^{\mathrm{Coded}}}{\mathrm{ACE}^{\mathrm{All}}}=\frac{\mathrm{ACE}^{\mathrm{Coded}}}{\mathrm{ACE}^{\mathrm{Coded}}+\mathrm{ACE}^{\mathrm{Uncoded}}} \tag{29}$$

By Eqs. (26) and (29), we obtain the effective bottom and top limits of the actual percentage of accumulated collage error, EBL-APACE and ETL-APACE, as

$$\underbrace{\frac{\mathrm{ACE}^{\mathrm{Coded}}}{\mathrm{ACE}^{\mathrm{Coded}}+\mathrm{ACE}^{\mathrm{Uncoded}}_{\mathrm{Max\_Effective}}}}_{\text{EBL-APACE}} \le \underbrace{\frac{\mathrm{ACE}^{\mathrm{Coded}}}{\mathrm{ACE}^{\mathrm{Coded}}+\mathrm{ACE}^{\mathrm{Uncoded}}}}_{\text{APACE}} \le \underbrace{\frac{\mathrm{ACE}^{\mathrm{Coded}}}{\mathrm{ACE}^{\mathrm{Coded}}+\mathrm{ACE}^{\mathrm{Uncoded}}_{\mathrm{Min\_Effective}}}}_{\text{ETL-APACE}} \tag{30}$$

From Eq. (30), we know that ETL-APACE<1 and EBL-APACE<ETL-APACE. When EBL-APACE reaches a large percentage (Such as 90%), EBL-APACE can approach ETL-APACE sufficiently, which implies that APACE lies in a rather small dynamic range and can be approximately estimated as

$$\mathrm{APACE}^{\mathrm{Estimated}} \approx \frac{\text{EBL-APACE}+\text{ETL - APACE}}{2} \tag{31}$$

Moreover, the average collage error (ACER) of all range blocks can be computed as

$$\mathrm{ACER}=\frac{\mathrm{ACE}^{\mathrm{All}}}{\mathrm{NumR}}=\sum_{i=1}^{\mathrm{NumR}}\mathrm{CE}(\boldsymbol{R}_i)\Big/\mathrm{NumR} \tag{32}$$

By Eqs. (32), (29), and (31), we can estimate ACER as

$$\mathrm{ACER}\approx\frac{\mathrm{ACE}^{\mathrm{Coded}}\big/\mathrm{APACE}^{\mathrm{Estimated}}}{\mathrm{NumR}} \tag{33}$$

The numerator of Eq. (33) represents the estimated version of $\mathrm{ACE}^{\mathrm{All}}$. In addition, the peak signal-to-noise ratio (PSNR) is used to measure the decoded image quality as

$$\mathrm{PSNR}=\log_{10}\left\{255^2\Big/\frac{1}{M\times N}\sum_{j=1}^{N}\sum_{i=1}^{M}\left[\boldsymbol{f}(i,j)-\boldsymbol{f}^{\mathrm{Decoded}}(i,j)\right]^2\right\} \tag{34}$$

where $\boldsymbol{f}$ and $\boldsymbol{f}^{\mathrm{Decoded}}$ represent the input and decoded images, respectively. Moreover, there exists a logarithmic relationship between ACER and the PSNR of the decoded image as[35]

$$\mathrm{PSNR}=\beta_1+\beta_2\log_{10}(\mathrm{ACER}) \tag{35}$$

where β_1 and β_2 are both constant values. Finally, by Eqs. (33) and (35), we can directly predict the PSNR of the decoded image.

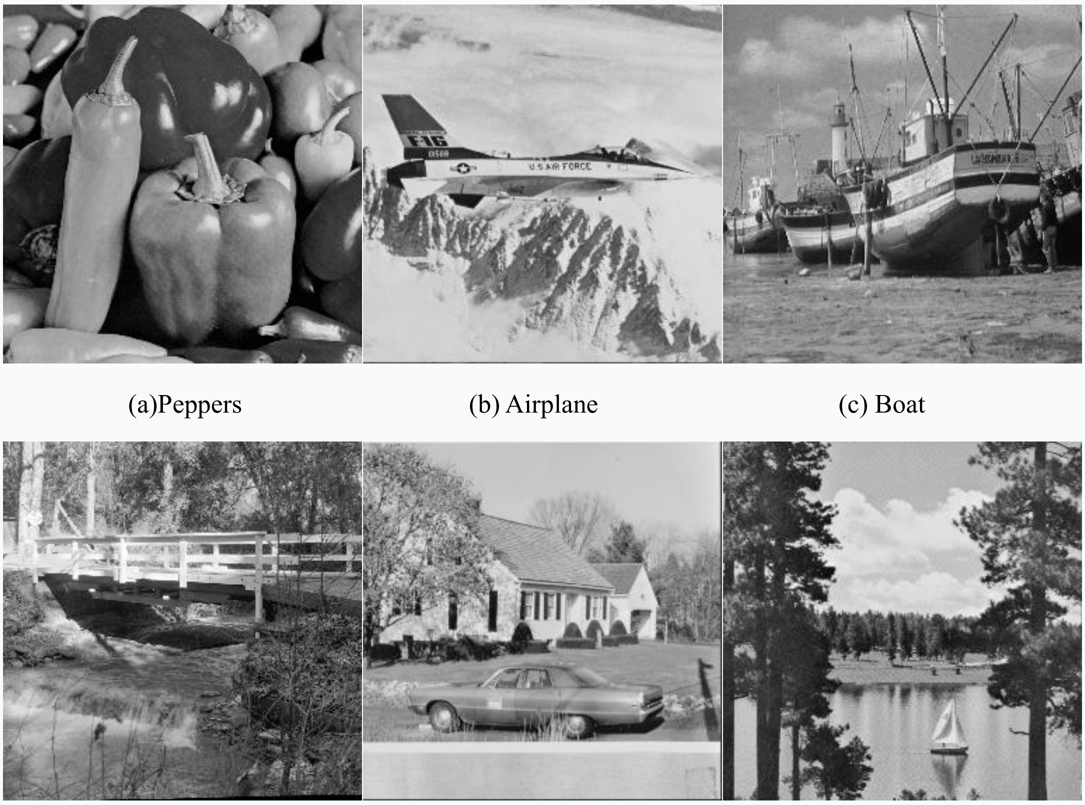
4. Experiments

We adopted six 256 × 256 images (in Fig.7), Peppers, Airplane, Boat, Bridge, House, and Lake, in our experiment. Four fractal image coding methods were used to assess the performance of the proposed method[1,5-7]. The scaling and offset coefficients, s and o, were quantized by 5 and 7 bits, respectively. The procedures of the proposed method are listed as follows:

Step 1: Divide the input image into range and domain blocks, and sort the range blocks by their variances from largest to smallest.

Step 2: Take out one uncoded range block and encode it. If EBL-APACE is smaller than 90%, turn back to Step 2. If not, calculate ETL-APACE and go to Step 3.

Step 3: Calculate ACER by Eq. (33), and predict the decoded image quality by Eq. (35).



(a)Peppers (b) Airplane (c) Boat

(d) Bridge (e) House (f) Lake

Fig.7 Six test images.

When the size of range blocks is set to be 4 × 4 and 8 × 8, respectively, Table 1, 2, 3, and 4 list the experimental results of Jacquin's, Chaurasia's, Zheng's, and Gupta's methods, respectively[1,5-7]. In Tab.1, when the size of range blocks is 4 × 4, for Jacquin's method, the second row shows the PSNRs of decoded images. In the third row, we encode all the range blocks, conduct the block-matching 4,096 times for all test images, and the percentage of computations is considered as 100%. For the previous method, the fourth to sixth rows list the predicted PSNRs, deviations between the predicted and actual PSNRs, and the percentage of computations with respect to Jacquin's method, respectively. Similarly, the seventh to ninth rows show the counterparts for the proposed method. From Tab.1, we see that if the size of range blocks is 4 × 4, the average deviations of the previous and proposed methods are 0.08dB and 0.05dB, respectively, the average percentages of computations are 52.99% and 47.76%, respectively. If the size of range blocks is 8 × 8, the average deviations of the previous and proposed methods are 0.10dB and 0.09dB, respectively, the percentage of computations needed are 52.62% and 50.73%, respectively. We see that compared with the previous method, the proposed method can provide higher prediction accuracy with fewer computations. The analyses are as follows: In previous method, $\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}$ and $\mathrm{LCC}_{\mathrm{Max}}^{\mathrm{Coded}}$ were used to approximate $\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Uncoded}}$ and $\mathrm{LCC}_{\mathrm{Max}}^{\mathrm{Uncoded}}$ and provided the bottom and top limits of APACE as

$$\mathrm{APACE}_{\mathrm{Min}}=\frac{\mathrm{ACE}^{\mathrm{Coded}}}{\mathrm{ACE}^{\mathrm{Coded}}+\mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Uncoded}}},\quad \mathrm{APACE}_{\mathrm{Max}}=\frac{\mathrm{ACE}^{\mathrm{Coded}}}{\mathrm{ACE}^{\mathrm{Coded}}+\mathrm{ACE}_{\mathrm{Min}}^{\mathrm{Uncoded}}} \tag{36}$$

where

$$\begin{cases}\mathrm{ACE}_{\mathrm{Max}}^{\mathrm{Uncoded}}=\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}}\mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Max}}=\frac{1}{B^2}\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}}\left\|\boldsymbol{R}_i^{\mathrm{Uncoded}}-\overline{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}\right)^2\right]\\ \mathrm{ACE}_{\mathrm{Min}}^{\mathrm{Uncoded}}=\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}}\mathrm{CE}(\boldsymbol{R}_i^{\mathrm{Uncoded}})_{\mathrm{Min}}=\frac{1}{B^2}\sum_{i=1}^{\mathrm{Num}^{\mathrm{Uncoded}}}\left\|\boldsymbol{R}_i^{\mathrm{Uncoded}}-\overline{r}_i\boldsymbol{I}\right\|^2\left[1-\left(\mathrm{LCC}_{\mathrm{Max}}^{\mathrm{Coded}}\right)^2\right]\end{cases} \tag{37}$$

Similar with Eq. (12), for coded range blocks, we also have

$$\begin{cases}\mathrm{LCC}_{\mathrm{Min\_Effective}}^{\mathrm{Coded}}>\mathrm{LCC}_{\mathrm{Min}}^{\mathrm{Coded}}\\ \mathrm{LCC}_{\mathrm{Max\_Effective}}^{\mathrm{Coded}}<\mathrm{LCC}_{\mathrm{Max}}^{\mathrm{Coded}}\end{cases} \tag{38}$$

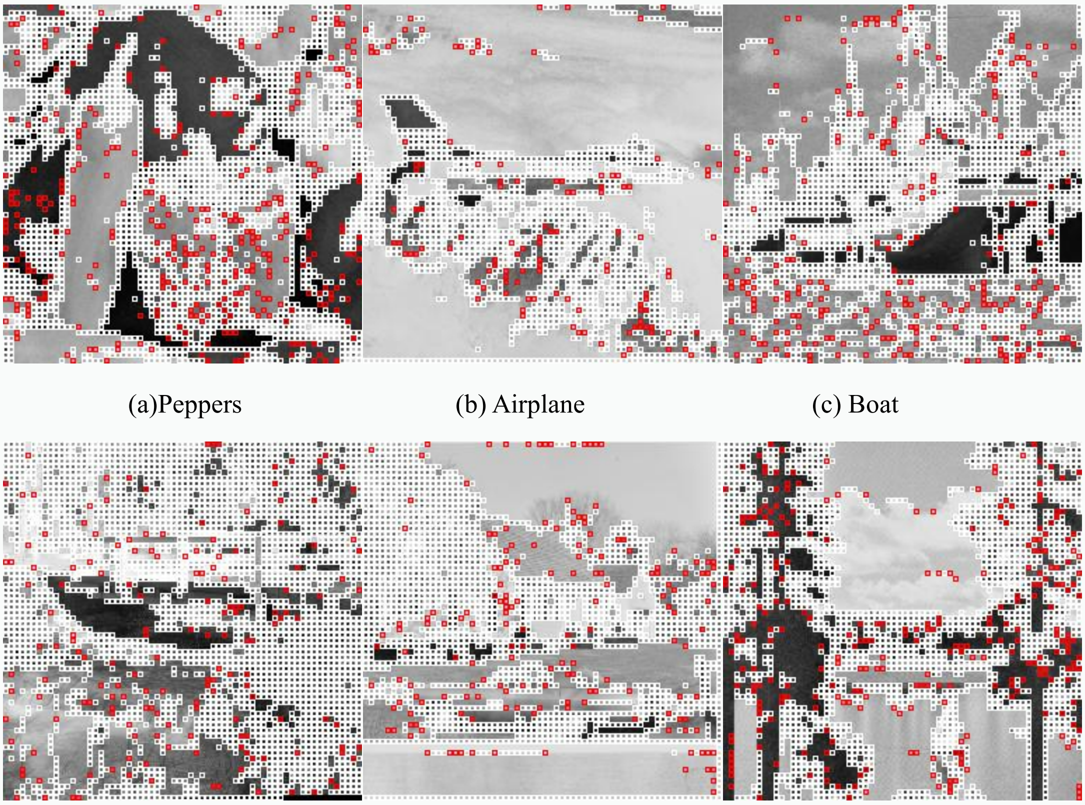
Then, by comparing Eq. (28) and (37), we have

$$\mathrm{APACE}_{\mathrm{Min}}<\text{EBL-APACE},\quad \text{ETL-APACE}<\mathrm{APACE}_{\mathrm{Max}} \tag{39}$$

By Eq. (39), $\mathrm{APACE}_{\mathrm{Min}}$ is always smaller than EBL-APACE. When EBL-APACE reaches

90% in the proposed method, $\text{APACE}_{\text{Min}}$ is smaller than 90%, which implies that in the previous method, we need to encode extra range blocks to make $\text{APACE}_{\text{Min}}$ reach 90%. Thus, a number of computations in the proposed method are saved. When the size of range blocks is set at 4 × 4, we totally have 4,096 range blocks. For Peppers, in the previous method, when $\text{APACE}_{\text{Min}}$ reaches 90%, we need to encode 2,327 range blocks which are represented by the red and white boxes in Fig.8 (a) and comprise 56.81% of total range blocks. In the proposed method, when EBL-APACE reaches 90%, we only need to encode 2,002 range blocks which are represented by white boxes in Fig.8 (a) and comprise 48.88% of total range blocks. This implies that the computations of 325 range blocks which are represented by red boxes are saved. When the size of range blocks is set at 8 × 8, we totally have 1,024 range blocks. In the previous method, when $\text{APACE}_{\text{Min}}$ reaches 90%, we need to encode 529 range blocks which are represented by the red and white boxes in Fig.9 (a) and comprise 51.66% of total range blocks. In the proposed method, when EBL-APACE reaches 90%, we need to encode 487 range blocks which are represented by the white boxes in Fig.9 (a) and comprise 47.56% of total range blocks. This implies that the computations of 42 range blocks which are represented by red boxes are saved. Moreover, By Eq. (39), we also know that compared with $\text{APACE}_{\text{Min}}$ to $\text{APACE}_{\text{Max}}$, EBL-APACE to ETL-APACE provides smaller dynamic range for estimating APACE, and the proposed method is expected to provide higher prediction accuracy.

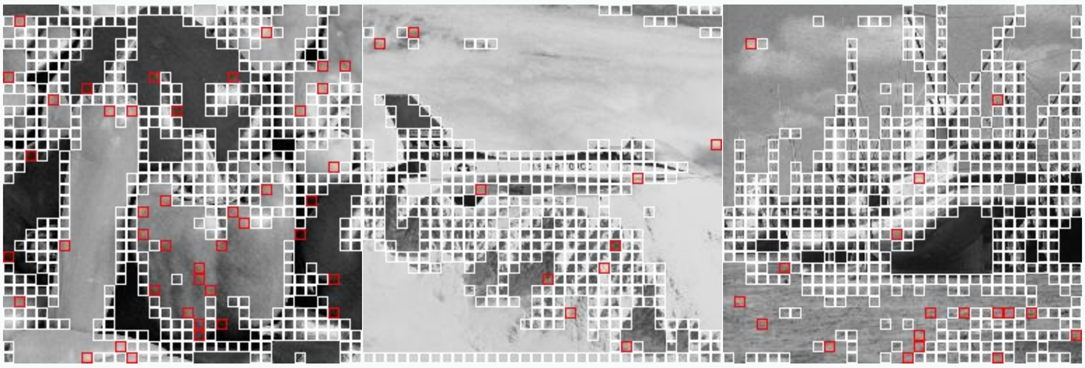
In Tab.2, for Chaurasia's method, if the size of range blocks is 4 × 4, the average deviations of the previous and proposed methods are both 0.07dB and 0.05dB, respectively. The average percentages of computations are 54.14% and 48.12%, respectively. If the size of range blocks is 8 × 8, the average deviations of the previous and proposed methods are both 0.17dB. The average percentages of computations are 52.44% and 50.31%, respectively. In Tab.3, for Zheng's method, if the size of range blocks is 4 × 4, the average deviations of the previous and proposed methods are 0.14dB and 0.10dB, respectively. The average percentages of computations are 53.87% and 47.70%, respectively. If the size of range blocks is 8 × 8, the average deviations of the previous and proposed methods are 0.11dB and 0.09dB, respectively, The average percentages of computations are 52.60% and 50.68%, respectively. In Tab.4, for Gupta's method, if the size of range blocks is 4 × 4, the average deviations of the previous and proposed methods are 0.08dB and 0.04dB, respectively. The average percentages of computations are 53.24% and 48.13%, respectively. If the size of range blocks is 8 × 8, the average deviations of the previous and proposed methods are 0.15dB and 0.14dB, respectively. The average percentages of computations are 52.80% and 50.90%, respectively. In summary, compared with the previous method, the proposed method can predict the decoded image quality more accurately with fewer computations.



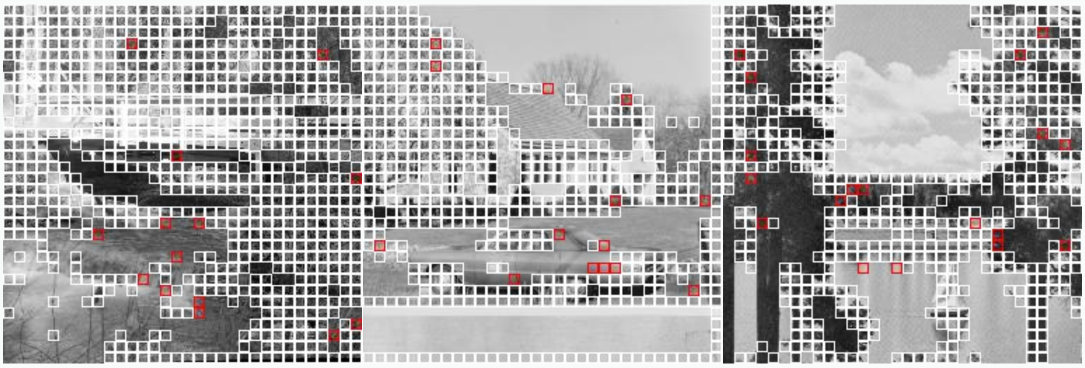
(a)Peppers (b) Airplane (c) Boat

(d)Bridge (e) House (f) Lake

Fig.8 When the size of range blocks is 4 × 4, white and red boxes represent the range blocks encoded in the previous method, and white boxes represent the range blocks encoded in the proposed method.



(a)Peppers (b) Airplane (c) Boat



(d)Bridge (e) House (f) Lake

Fig.9 When the size of range blocks is 8 × 8, white and red boxes represent the range blocks encoded in the previous method, and white boxes represent the range blocks encoded in the proposed method.

Tab.1 Performance comparison between the previous and proposed methods for Jacquin's method[1].

Test images			Peppers	Airplane	Boat	Bridge	House	Lake	Average
4 × 4	Jacquin's[1]	Decoded(dB)	31.80	29.46	28.28	25.44	27.56	27.72	×
		Computations(%)	100	100	100	100	100	100	×
	Previous[35]	Predicted(dB)	31.64	29.39	28.30	25.55	27.52	27.64	×
		Deviation(dB)	**0.16**	**0.07**	**0.02**	**0.11**	**0.04**	**0.08**	**0.08**
		Computations(%)	**56.81**	**35.38**	**53.93**	**71.17**	**50.27**	**50.37**	**52.99**
	Proposed	Predicted(dB)	31.71	29.42	28.32	25.56	27.55	27.71	×
		Deviation(dB)	**0.09**	**0.04**	**0.04**	**0.12**	**0.01**	**0.01**	**0.05**
		Computations(%)	**48.88**	**32.50**	**48.24**	**67.11**	**46.17**	**43.65**	**47.76**
8 × 8	Jacquin's[1]	Decoded(dB)	25.28	23.17	23.28	21.19	22.37	22.18	×
		Computations(%)	100	100	100	100	100	100	×
	Previous[35]	Predicted(dB)	25.27	23.14	23.17	21.11	22.52	21.93	×
		Deviation(dB)	**0.01**	**0.03**	**0.11**	**0.08**	**0.15**	**0.25**	**0.11**
		Computations(%)	**51.66**	**36.91**	**53.71**	**72.17**	**51.37**	**49.90**	**52.62**
	Proposed	Predicted(dB)	25.28	23.11	23.16	21.07	22.48	21.92	×
		Deviation(dB)	**0.00**	**0.06**	**0.12**	**0.12**	**0.11**	**0.16**	**0.10**
		Computations(%)	**47.56**	**35.94**	**51.95**	**70.80**	**50.00**	**48.14**	**50.73**

Tab.2 Performance comparison between the previous and proposed methods for Chaurasia's method[5].

Test images			Peppers	Airplane	Boat	Bridge	House	Lake	Average
4 × 4	Chaurasia's[5]	Decoded(dB)	31.47	29.37	28.04	25.16	27.08	27.28	×
		Computations(%)	100	100	100	100	100	100	×
	Previous[35]	Predicted(dB)	31.30	29.23	28.00	25.12	27.07	27.26	×
		Deviation(dB)	**0.17**	**0.14**	**0.04**	**0.04**	**0.01**	**0.02**	**0.07**
		Computations(%)	**59.72**	**35.25**	**55.98**	**72.02**	**50.10**	**51.76**	**54.14**
	Proposed	Predicted(dB)	31.40	29.26	28.05	25.14	27.09	27.34	×
		Deviation(dB)	**0.07**	**0.11**	**0.01**	**0.02**	**0.01**	**0.06**	**0.05**
		Computations(%)	**49.37**	**32.74**	**48.75**	**67.77**	**46.17**	**43.92**	**48.12**

8 × 8	Chaurasia's[5]	Decoded(dB)	24.97	23.15	23.08	20.64	22.19	21.82	×
		Computations(%)	100	100	100	100	100	100	×
	Previous[35]	Predicted(dB)	24.85	23.03	22.89	20.87	22.37	21.64	×
		Deviation(dB)	**0.12**	**0.12**	**0.19**	**0.23**	**0.18**	**0.18**	**0.17**
		Computations(%)	**50.39**	**36.91**	**54.30**	**72.07**	**51.07**	**49.90**	**52.44**
	Proposed	Predicted(dB)	24.86	23.00	22.90	20.85	22.33	21.61	×
		Deviation(dB)	**0.11**	**0.15**	**0.18**	**0.21**	**0.14**	**0.21**	**0.17**
		Computations(%)	**45.90**	**35.74**	**51.66**	**70.61**	**50.00**	**47.95**	**50.31**

Tab.3 Performance comparison between the previous and proposed methods for Zheng's method[6].

Test images			Peppers	Airplane	Boat	Bridge	House	Lake	Average
4 × 4	Zheng's[6]	Decoded(dB)	31.78	29.29	28.04	25.06	27.46	27.70	×
		Computations(%)	100	100	100	100	100	100	×
	Previous[35]	Predicted(dB)	31.61	29.19	28.07	25.43	27.43	27.56	×
		Deviation(dB)	**0.17**	**0.10**	**0.03**	**0.37**	**0.03**	**0.14**	**0.14**
		Computations(%)	**57.50**	**35.99**	**55.08**	**70.78**	**50.02**	**53.86**	**53.87**
	Proposed	Predicted(dB)	31.69	29.24	28.10	25.44	27.46	27.67	×
		Deviation(dB)	**0.09**	**0.05**	**0.06**	**0.38**	**0.00**	**0.03**	**0.10**
		Computations(%)	**48.83**	**32.37**	**48.90**	**66.65**	**45.92**	**43.55**	**47.70**
8 × 8	Zheng's[6]	Decoded(dB)	25.24	23.06	23.25	20.99	22.36	21.77	×
		Computations(%)	100	100	100	100	100	100	×
	Previous[35]	Predicted(dB)	25.26	23.14	23.14	21.11	22.51	21.93	×
		Deviation(dB)	**0.02**	**0.08**	**0.11**	**0.12**	**0.15**	**0.16**	**0.11**
		Computations(%)	**51.66**	**36.91**	**53.61**	**72.17**	**51.37**	**49.90**	**52.60**
	Proposed	Predicted(dB)	25.28	23.11	23.13	21.07	22.47	21.91	×
		Deviation(dB)	**0.04**	**0.05**	**0.12**	**0.08**	**0.11**	**0.14**	**0.09**
		Computations(%)	**47.56**	**35.84**	**51.76**	**70.80**	**50.00**	**48.14**	**50.68**

Tab.4 Performance comparison between the previous and proposed methods for Gupta's method[7].

Test images			Peppers	Airplane	Boat	Bridge	House	Lake	Average
4 × 4	Gupta's[7]	Decoded(dB)	31.75	29.29	28.16	25.32	27.48	27.66	×
		Computations(%)	100	100	100	100	100	100	×
	Previous[35]	Predicted(dB)	31.57	29.26	28.14	25.38	27.41	27.56	×
		Deviation(dB)	**0.18**	**0.03**	**0.02**	**0.06**	**0.07**	**0.10**	**0.08**
		Computations(%)	**58.20**	**35.69**	**54.39**	**71.41**	**50.66**	**49.07**	**53.24**
	Proposed	Predicted(dB)	31.66	29.29	28.15	25.38	27.43	27.61	×
		Deviation(dB)	**0.09**	**0.00**	**0.01**	**0.06**	**0.05**	**0.05**	**0.04**
		Computations(%)	**48.58**	**32.81**	**49.71**	**67.29**	**46.53**	**43.85**	**48.13**
8 × 8	Gupta's[7]	Decoded(dB)	25.26	23.08	23.22	20.88	22.20	22.13	×
		Computations(%)	100	100	100	100	100	100	×
	Previous[35]	Predicted(dB)	25.19	23.00	23.06	21.00	22.39	21.85	×
		Deviation(dB)	**0.07**	**0.08**	**0.16**	**0.12**	**0.19**	**0.28**	**0.15**
		Computations(%)	**51.56**	**37.30**	**54.20**	**72.36**	**51.17**	**50.20**	**52.80**

Proposed	Predicted(dB)	25.19	22.97	23.06	20.96	22.34	21.84	×
	Deviation(dB)	**0.07**	**0.11**	**0.16**	**0.08**	**0.14**	**0.29**	**0.14**
	Computations(%)	**48.34**	**36.04**	**51.76**	**70.70**	**50.10**	**48.44**	**50.90**

5. Conclusion

An effective method for predicting the fractal decoded image quality was proposed in this paper. First, we introduced the effective minimum and maximum of LCCs, which provided EBL-APACE and ETL-APACE for APACE. Then, decoded image quality can be directly predicted while EBL-APACE reached a large percentage. Experimental results show that compared with the previous method, the proposed method can provide higher prediction accuracy with fewer computations. In future work, we will focus our attention on the following two aspects:

a) Maintain prediction accuracy. By removing the interference of the outliers of LCCs, we have obtained satisfying prediction accuracy. In future work, we will try to maintain the current prediction accuracy.

b) Shorten prediction process. Compared with the previous method, the number of computations in the proposed method has been reduced. In future work, we will try to reduce the number of computations further.

Acknowledgments

This work was partially supported by the State Scholarship Fund of China (Grant No. 201806575003) and Special Project of Collaborative Innovation of Hengshui University (Grant No. 2022XJZX19).

References

1. A. K. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. on Image Process.* **1**(1), 18-30 (1992).
2. Y. Fisher, *Fractal image compression: Theory and Application* (Springer-Verlag, 1994).
3. B. Wohlberg and B. G. Jager, "A review of the fractal image coding literature," *IEEE Trans. on Image Process.* **8**(12), 1716-1729 (1999).

4. A. K. Jacquin, “Fractal image coding: a review,” *P. IEEE.* **81**(10), 1451-1465 (1993).

5. V. Chaurasia and V. Chaurasia, “Statistical feature extraction based technique for fast fractal image compression,” *J. Vis. Commun. Image R.* **41,** 87-95 (2016).

6. Y. P. Zheng, X. P. Li, and M. Sarem, “Fast fractal image compression algorithm using specific update search,” *IET Image Process.* **14**(9), 1733-1739 (2020).

7. R. Gupta, D. Mehrotra, and R. K. Tyagi, “Hybrid edge-based fractal image encoding using K-NN search,” *Multimed. Tools Appl.* **81,** 21135-21154 (2022).

8. F. R. Shen and H. Osamu, “A fast no search fractal image coding method,” *Signal Process. Image Commun.* **19**(5), 393-404 (2004).

9. X. Y. Wang and S. G. Wang, “An improved no-search fractal image coding method based on a modified gray-level transform,” *Comput. Graph.* **32**(4), 445-450 (2008).

10. X. Y. Wang, Y. X. Wang, and J. J. Yun, “An improved no-search fractal image coding method based on a fitting plane,” *Image Vision Comput.* **28,** 1303-1308 (2010).

11. S. Bi and Q. Wang, “Fractal image coding based on a fitting surface” *J. Appl. Math.* **2014,** 634848 (2014).

12. M. Ghazel, G. H. Freeman, and E. R. Vrscay, “Fractal image denoising,” *IEEE Trans. on Image Process.* **12**(12), 1560-1578 (2003).

13. M. Ghazel, G. H. Freeman, and E. R. Vrscay, “Fractal-wavelet image denoising revisited,” *IEEE Trans. on Image Process.* **15**(9), 2669-2675 (2006).

14. J. Lu, Z. X. Ye, Y. Y. Zou, and R. S. Ye, “An enhanced fractal image denoising algorithm,” *Chaos, Soliton. Fract.* **38,** 1054-1064 (2008).

15. J. H. Jeng, C. C. Tseng, and J. G. Hsieh, “Study on huber fractal image compression,” *IEEE Trans. on Image Process.* **18**(5), 995-1003 (2009).

16. J. Lu, Z. X. Ye, and Y. Y. Zou, “Huber fractal image coding based on a fitting plane,” *IEEE Trans. on Image Process.* **22**(1), 134-145 (2013).

17. Y. Y. Zou, H. X. Hu, J. Lu, X. X. Liu, Q. T. Jiang, and G. H. Song, “A nonlocal low-rank regularization method for fractal image coding,” *Fractals.* **29**(5), 2150125 (2021).

18. C. Xu, Y. T. Ye, and Z. W. Hu, “A primal-dual algorithm for robust fractal image coding,” *Fractals.* **27**(7), 1950119 (2019).

19. S. M. Abdullahi, H. X. Wang, and T. Li, “Fractal coding-based robust and alignment-free

fingerprint image hashing," *IEEE Trans. on Inf. Foren. Sec.* **15,** 2587-2601 (2020).

20. F. Khelaifi and H. J. He, "Perceptual image hashing based on structural fractal features of image coding and ring partition," *Multimed. Tools Appl.* **79**(27), 19025-19044 (2020).

21. M. H. Pi, M. K. Mandal, and A. Basu, "Image retrieval based on histogram of fractal parameters," *IEEE Trans. on Multimedia.* **7**(4), 597-605 (2005).

22. X. Y. Wang and Z. Chen, "A fast fractal coding in application of image retrieval," *Fractals.* **17**(4), 441-450 (2009).

23. X. Huang, Q. Zhang, and W. Liu, "A new method for image retrieval based on analyzing fractal coding characters," *J. Vis. Commun. Image R.* **24**(1), 42-47 (2013).

24. M. H. Pi, C. H. Li, and H. Li, "A novel fractal image watermarking," *IEEE Trans. on Multimedia.* **8**(3), 488-498 (2006).

25. F. Daraee and S. Mozaffari, "Watermarking in binary document images using fractal codes," *Pattern Recogn. Lett.* **35,** 120-129 (2014).

26. J. Lu, Y. Y. Zou, C. Y. Yang, and L. J. Wang, "A robust fractal color image watermarking algorithm," *Math. Probl. Eng.* **2014,** 638174 (2014).

27. K. H. Chung, Y. H. Fung, and Y. H. Chan, "Image enlargement using fractal," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* 273-275 (2003).

28. C. M. Lai, K. M. Lam, Y. H. Chan, and W. C. Siu, "An efficient fractal based algorithm for image magnification," in *Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing,* Hong Kong 571-574 (2004).

29. Z. P. Chen, Z. L. Ye, S. X. Wang, and G. H. Peng, "Image magnification based on similarity analogy," *Chaos, Soliton. Fract.* **40,** 2370-2375 (2009).

30. Y. C. Wee and H. J. Shin, "A novel fast fractal super resolution technique," *IEEE Trans. Consum. Electron.* **56**(3), 1537-1541 (2010).

31. Z. Hua, H. C. Zhang, and J. J. Li, "Image super resolution using fractal coding and residual network," *Complexity.* **2019,** 9419107 (2019).

32. C. Bisogni, M. Nappi, C. Pero, and S. Ricciardi, "FASHE: A fractal based strategy for head pose estimation," *IEEE Trans. on Image Process.* **30**, 3192-3203 (2021).

33. C. Bisogni, M. Nappi, C. Pero, and S. Ricciardi, "PIFS scheme for head pose estimation aimed at faster face recognition," *IEEE Trans. on Biometrics, Behavior, and Identity Science* **4**, 173-184

(2022).

34. S. Liu, W. L. Bai, N. Y. Zeng, and S. H. Wang, “A fast fractal based compression for MRI images,” *IEEE Access.* **7,** 62412-62420 (2019).

35. Q. Wang and S. Bi, “Improved method for predicting the peak signal-to-noise ratio quality of decoded images in fractal image coding,” *J. Electron. Imaging* **26**(1), 013024 (2017).